Research Report Number 71-6
Department of Computer Science
YALE UNIVERSITY
New Haven, Connecticut 06520
March 1971

NEW COMPUTATIONAL ALGORITHMS
FOR MINIMIZING A SUM OF SQUARES
OF NONLINEAR FUNCTIONS

by

Kenneth M. Brown and J. E. Dennis, Jr.

†Department of Computer Science
Cornell University
Ithaca, New York 14850
NEW COMPUTATIONAL ALGORITHMS FOR MINIMIZING
A SUM OF SQUARES OF NONLINEAR FUNCTIONS

KENNETH M. BROWN† AND J. E. DENNIS, JR.‡

Abstract. We introduce two new computational algorithms for minimizing a sum of squares, $\phi$, of nonlinear functions. (The methods can, of course, be used for the important special case of nonlinear least squares curve fitting.) The algorithms are closely related to Newton's method for finding a zero of the gradient of $\phi$; however, we are able to avoid the explicit calculation of the second partial derivatives. Local and Kantorovich-type convergence theorems are proven for the algorithms. The results of computational experiments are presented, including cases in which $\phi$ is quite large at the minimum point.

Introduction. Let $x \in E^N$ and suppose we wish to minimize

$$\phi(x) = \sum_{i=1}^{M} f_i(x)^2,$$

where the $f_i$'s are nonlinear functions of $x$. Now every such relative

† Department of Computer Science, Yale University, New Haven, Connecticut 06520. Now at Department of Computer, Information, and Control Sciences, University of Minnesota, Minneapolis, Minnesota 55455. The work of this author was supported in part by the Office of Naval Research under Contract NR 044-401.

‡ Department of Computer Science, Cornell University, Ithaca, New York 14850. The work of this author was supported in part by the National Science Foundation under Grant GJ-844.
minimum will be found among the zeros of \( V\phi(x) \), the gradient of \( \phi \).

Call such a zero \( x^* \). Now

\[
V\phi(x) = 2 J_F^T(x)F(x),
\]

where \( J_F(x) \) denotes the \( M \times N \) Jacobian matrix of \( F = (f_1, \ldots, f_M)^T \). Define

\[
G(x) = J_F^T(x)F(x).
\]

We seek the zeros of \( G(x) \) and hence of the gradient \( V\phi(x) \).

Let \( H_k(x) \) denote the Hessian matrix of \( f_k \) at \( x \); i.e.,

the \( i^{th}, j^{th} \) element of \( H_k(x) \) is given by \( \partial^2 f_k(x)/\partial x_i \partial x_j \). By direct calculation we have that

\[
(1) \quad J_G(x) = \sum_{k=1}^{M} f_k(x)H_k(x) + J_F(x)^TJ_F(x),
\]

so that Newton's method applied to

\[
(2) \quad G(x) = 0
\]

is given by
NEW ALGORITHMS FOR MINIMIZING A SUM OF SQUARES

\[ (3) \quad x_{n+1} = x_n - \left[ \sum_{k=1}^{M} f_k(x_n) H_k(x_n) + J_F(x_n) J_F(x_n)^T \right]^{-1} J_F(x_n) F(x_n). \]

(See, for example, [18, p. 269].)

The latter formula requires (assuming continuous second partial derivatives of \( \phi \)) the calculation of \( M \cdot N \cdot (N + 1)/2 \) second partial derivatives per iterative step. The Gauss [12] and Levenberg-Marquardt [14], [15] algorithms are two frequently used attempts to circumvent this difficulty.

The former simply drops the term \( \sum_{k=1}^{M} f_k(x_n) H_k(x_n) \) and the latter approximates it with a diagonal matrix \( \mu_n I \). Both methods work well locally when \( ||F(x)|| \) is very small at a zero of \( G \). For example in [4], we have shown that the Levenberg-Marquardt iteration converges quadratically to \( x^* \), a zero of \( F \), if \( \mu_n = 0( ||F(x_n)|| ) \) and \( J_F(x^*) \) has full rank. Obviously the Gauss method behaves likewise. We have also shown [4] identical results for the derivative-free analogues of these methods. When \( \phi(x^*) \) is large, the Levenberg-Marquardt algorithm can degenerate into an awkward descent method, for then the \( H_k \) in (3) are no longer damped out.

In §2 we introduce two algorithms based on approximating the \( H_k \) in (3) without requiring additional function or derivative evaluations. In §3, we prove local and Kantorovich-type convergence theorems for the algorithms. The results of numerical experiments are presented in §4.
In the latter section we treat 1) a well studied example so as to compare our methods with other currently used minimization techniques, 2) an application of these methods to the problem of determining the weights and nodes of quadrature formulas and 3) a problem having an extremely large residual at the minimum.

A preliminary statement of a portion of these results was given without proofs in [5].
2. Description of the algorithms. In order to approximate the $H_k$ without requiring additional derivative or function evaluations, we propose the following algorithms.

ALGORITHM 2.1. Let $x_n$, $J_F(x_n)$ and $F(x_n)$ be given along with $M$ matrices $B_{1,n}, \ldots, B_{M,n}$ each of size $N \times N$. (Initially the $B_{i,o}$ may be chosen to approximate the $H_{i,o}$ by, say, using first differences on the entries of $J_F(x)$; this technique was used in the numerical experiments described in §4.) Obtain

\[
x_{n+1} = x_n - \left[ \sum_{k=1}^{M} f_k(x_n) B_{k,n} + J_F(x_n)^T J_F(x_n) \right]^{-1} J_F(x_n)^T F(x_n)
\]

\[
= x_n - A_{n}^{-1} J_F(x_n)^T F(x_n)
\]

and compute $J_F(x_{n+1})$ and $F(x_{n+1})$. Now update the $B_{i}$ by means of

\[
B_{i,n+1} = B_{i,n} + [\nabla f_i(x_{n+1})^T - \nabla f_i(x_n)^T]
\]

\[
- B_{i,n} (x_{n+1} - x_n) \frac{(x_{n+1} - x_n)^T}{\| x_{n+1} - x_n \|^2_2}
\]

for each $i = 1, \ldots, M$. Continue the process until termination criteria are met.
ALGORITHM 2.2. This algorithm is exactly the same as Algorithm 2.1 except that $B_{1,n}, \ldots, B_{M,n}$ are initialized as symmetric matrices and in place of (5) one generates the sequence of approximate Hessians by

\begin{equation}
B_{i,n+1} = B_{i,n} + [\nabla f_i(x_{n+1})^T - \nabla f_i(x_n)^T - B_{i,n}(x_{n+1} - x_n)] \frac{(x_{n+1} - x_n)^T}{||x_{n+1} - x_n||_2^2} \frac{(x_{n+1} - x_n)}{||x_{n+1} - x_n||_2^2} \\
+ \frac{(x_{n+1} - x_n)^T}{||x_{n+1} - x_n||_2^2} [\nabla f_i(x_{n+1})^T - \nabla f_i(x_n)^T - B_{i,n}(x_{n+1} - x_n)]^T \\
- (x_{n+1} - x_n)[\nabla f_i(x_{n+1})^T - \nabla f_i(x_n)^T - B_{i,n}(x_{n+1} - x_n)]^T (x_{n+1} - x_n) \frac{(x_{n+1} - x_n)^T}{||x_{n+1} - x_n||_2^4}.
\end{equation}

REMARK 2.1. $\nabla f_i(x)$ is just the $i$th row of $J_f(x)$. 

REMARK 2.2. Equations (5) and (6) are respectively the appropriate generalizations of Broyden's "single-rank" approximation [6] to $H_i(x_n)$ and Powell's symmetric form of Broyden's approximation [21].
REMARK 2.3. These algorithms require no more function or derivative evaluations than do the Gauss [4,14] or Levenberg–Marquardt [4,15] algorithms; however, more storage space is needed. The additional storage requirement is offset on the one hand by superior local behavior (stability near a root) and on the other hand by a gain in speed of convergence. Since the $B_i$'s generated by (6) are symmetric Algorithm 2.2 will require less storage than Algorithm 2.1.
Convergence results. The purpose of this section is to present theorems which characterize the main convergence properties of the algorithms.

The following lemma bounds the error in the Hessian approximations given by (5).

**Lemma 1.** Let \( \Omega \) be an open convex neighborhood of \( x^* \), and let \( K > 0 \) be a constant and \( P \) be a Frechet differentiable function mapping \( \Omega \) into \( E^N \) such that for every \( x \in \Omega \)

\[
(L1) \quad \| J_p(x^*) - J_p(x) \| \leq K \| x^* - x \| .
\]

Let \( B \) be a real \( N \times N \) matrix and let \( x, x' \in \Omega \). Define \( B' \) by

\[
(5') \quad B' \equiv B + [P(x') - P(x) - B(x' - x)] \frac{(x' - x)^T}{\| x' - x \|^2} .
\]

Under these hypotheses

\[
\| B' - J_p(x') \| \leq \| B - J_p(x) \| + 2K(\| x - x^* \| + \| x' - x^* \|) .
\]

If, in addition,

\[
(L2) \quad \| J_p(x) - J_p(y) \| \leq K \| x - y \| ,
\]
then

\[ ||B' - J_p(x')|| \leq ||B - J_p(x)|| + \frac{3}{2} k ||x' - x||. \]

**Proof.** Now

\[ B' - J_p(x') = B - J_p(x) + [P(x') - P(x) - B(x' - x)] \frac{(x' - x)^T}{||x' - x||^2} + J_p(x) - J_p(x') \]

\[ = B - J_p(x) + [P(x') - P(x) - J_p(x)(x' - x)] \frac{(x' - x)^T}{||x' - x||^2} \]

\[ + [J_p(x) - B] \frac{(x' - x)(x' - x)^T}{||x' - x||^2} + J_p(x) - J_p(x') \; ; \; \text{thus} \]

\[ ||B' - J_p(x')|| \leq ||B - J_p(x)|| \cdot ||I - \frac{(x' - x)(x' - x)^T}{||x' - x||^2}|| \]

\[ + ||P(x') - P(x) - J_p(x)(x' - x)|| \cdot ||x' - x||^{-1} + ||J_p(x) - J_p(x')|| \cdot \]

Now from [6], \[||I - \frac{(x' - x)(x' - x)^T}{||x' - x||^2}|| = 1 . \] If \( J_p \) is Lipschitz (L2 holds), the corresponding result is clear. If we assume only the one sided Lipschitz condition \( LL \), then the above reduces to
\[ ||B' - J_p(x')|| \leq ||B - J_p(x)|| + \sup_{\xi \in (x',x)} ||J_p(x) - J_p(\xi)|| + ||J_p(x) - J_p(x')|| \]
\[ \leq ||B - J_p(x)|| + 2K(||x - x^*|| + ||x' - x^*||) \]
by adding and subtracting \( J_p(x^*) \) twice.

**Lemma 2.** Let hypothesis \( L_1 \) of Lemma 1 hold but let \( B \) be a real \( N \times N \) symmetric matrix and define \( B' \) by

\[ (6') \quad B' \equiv B + [P(x') - P(x) - B(x' - x)] \frac{(x' - x)^T}{||x' - x||^2} \]
\[ + \frac{(x' - x)}{||x' - x||^2} \frac{[P(x') - P(x) - B(x' - x)]^T}{||x' - x||^4} \]
\[ - (x' - x) \frac{[P(x') - P(x) - B(x' - x)]^T}{||x' - x||^4} (x' - x) \frac{(x' - x)^T}{||x' - x||^4} \]

Under these hypotheses,

\[ ||B' - J_p(x')|| \leq ||B - J_p(x)|| + 3K (||x - x^*|| + ||x' - x^*||) \].
If L2 holds then

$$||B' - J_F(x')|| \leq ||B - J_F(x)|| + 2K||x' - x||.$$ 

**Lemma 3.** If, for each $i = 1, \ldots, M$, $H_i$ satisfies L1 with constant $K_i$ on a compact convex subset $C$ of $\Omega$ then there is a constant $\gamma_1$ such that $J_G$ satisfies L1 with $K = \gamma_1$ on $C$.

**Proof.** Let $K = \max K_i$ and select $B \geq ||H_i(x)||$, $B' \geq ||J_F(x)|| = ||J_F(x)^T||$ (see [25]), $B'' \geq ||F(x)||_1$ for every $x \in C$, $i = 1, \ldots, M$. These constants can be chosen because of the continuity of every $H_i$ and the compactness of $C$. Notice that $M^{1/2}B$, $B'$ serve as Lipschitz constants for $J_F$ and $F$ on $C$.

Assume that every $H_i$ satisfies L1, then

$$||J_G(x) - J_G(x^*)|| = ||\Sigma f_i(x)H_i(x) + J_F^T(x)J_F(x) - \Sigma f_i(x^*)H_i(x^*) - J_F^T(x^*)J_F(x^*)||$$

$$\leq ||\Sigma [f_i(x) - f_i(x^*)]H_i(x)|| + ||\Sigma f_i(x^*)[H_i(x) - H_i(x^*)]||$$

$$+ ||J_F(x)^T[J_F(x) - J_F(x^*)]|| + ||[J_F(x)^T - J_F(x^*)^T]J_F(x^*)||$$
\[
\leq \frac{M}{M} \left( \sum_{i=1}^{M} \left\| H_i(x) \right\|^2 \right)^{1/2} \\
+ \left\| F(x^*) \right\|_1 \cdot \max_{i} \left\| H_i(x) - H_i(x^*) \right\| \\
+ \left\| J_F(x) \right\| \cdot \left\| J_F(x) - J_F(x^*) \right\| \\
+ \left\| J_F(x^*) \right\| \cdot \left\| J_F(x^*)^T - J_F(x^*)^T \right\| \\
\leq (3M^{1/2}B + B^nK) \left\| x - x^* \right\| \\
\equiv \gamma_1 \left\| x - x^* \right\|
\]

REMARK 3.1. The lemma and its proof are exactly the same if we replace \( L_1 \) by \( L_2 \).

THEOREM 1. Let \( x^* \) be a zero of \( J_F^T(\cdot)F(\cdot) \) and let \( K_1 > 0 \) be constants such that for every \( x \in \Omega \), \( \left\| H_i(x) - H_i(x^*) \right\| \leq K_1 \left\| x - x^* \right\| \) for each \( i = 1, \ldots, M \). Whenever \( J_G(x^*) \) is non-singular, there exist constants \( \delta > 0, \varepsilon > 0 \) such that if \( \left\| x_0 - x^* \right\| < \varepsilon \) and \( \left\| B_{i,0} - H_i(x_0) \right\| \leq \delta \), \( i = 1, \ldots, M \), then Algorithms 2.1 and 2.2 converge to \( x^* \) from \( x_0 \).
Proof. Choose $C$ to be the closure of a conditionally compact convex neighborhood of $x^*$. Furthermore, choose $C$ sufficiently small so that $J_G$ is invertible on $C$ and $||J_G(x)^{-1}||$ is uniformly bounded by some constant $B$. Select a constant $B'$ such that $||F(x)||_1$ is uniformly bounded on $C$ by $B'$. Let $K = \max_i K_i$ and pick $\delta < (6BB')^{-1}$ and $\epsilon \leq \max \left( \frac{B'\delta}{\gamma_1}, \frac{\delta}{9K} \right)$ such that $N(x^*, \epsilon) \subset C$, where $\gamma_1$ is the constant for which $J_G$ satisfies $L_1$ on $C$ from Lemma 3. Now select $x_{o}, B_{1,o}, \ldots, B_{M,o}$ as above. Set $A_o = \sum_{i=1}^{M} f_i(x_o)B_{i,o} + J_{F}(x_o)^TJ_F(x_o)$. Now

$$||A_o - J_G(x_o)|| = \left|\left| \sum_{i=1}^{M} f_i(x_o)B_{i,o} - H_i(x_o) \right|\right|$$

$$\leq \sum_{i=1}^{M} |f_i(x_o)| \cdot ||B_{i,o} - H_i(x_o)||$$

$$\leq ||F(x_o)||_1 \delta$$

$$\leq B'\delta.$$ 

Hence $||I - J_G(x_o)^{-1}A_o|| \leq BB'\delta < 1$ and so $A_o^{-1}$ exists and is bounded in norm by $B(1 - BB'\delta)^{-1}$. Thus $x_1 = x_o - A_o^{-1}G(x_o)$ exists.
Set $e_1 = ||x_1 - x^*||$. Now

$$e_1 \leq ||A_o^{-1}|| \cdot \left( ||G(x^*) - G(x_o) - A_o(x^* - x_o)|| \right)$$

$$\leq ||A_o^{-1}|| \cdot \left( ||G(x^*) - G(x_o) - J_G(x_o)(x^* - x_o)|| + ||J_G(x_o) - A_o||e_o \right)$$

$$\leq B(1 - BB')^{-1}[Y_1 e_o + ||F(x_o)||_1 e_o] e_o$$

$$\leq \frac{BY_1 e_o + BB'\delta}{1 - BB'\delta} e_o \leq \frac{2BB'\delta}{1 - BB'\delta} e_o \leq \frac{2}{5} \cdot \frac{8}{5} e_o < \frac{1}{2} e_o.$$

Hence $J_G(x_1)$ and $A_1$ exist and as before:

$$||A_1 - J_G(x_1)|| \leq ||F(x_1)||_1 \max_i ||B_{i,1} - H_i(x_1)||$$

$$\leq B'\left[\delta + 3K(e_o + e_1)\right] \leq B'(\delta + \frac{9}{2}Ke_o) \leq B'\delta \cdot \frac{3}{2}.$$

Notice that we have used the bounds for Algorithm 2.2 so that the proof will also work for Algorithm 2.1 since the bound in that case is smaller. Thus, $||I - J_G(x_1)^{-1}A_1|| \leq BB'\delta \cdot 3/2 < 2BB'\delta < 1$. This means $A_1^{-1}$ exists and is bounded in norm by $B(1 - BB'\delta)$, so $x_2$ exists.

Assume by way of induction that $x_1, \ldots, x_n$, $A_1^{-1}, \ldots, A_{n-1}^{-1}$ all exist, that $e_k \leq \frac{1}{2} e_{k-1}$ and that $\max_i ||B_{i,k} - H_i(x_k)|| \leq (2 - \left(\frac{1}{2}\right)^k)\delta$, $k \leq n$. 
Then \[ \| A_n - J_G(x_n) \| \leq \| F(x_n) \|_1 \max_i \| B_{i,n} - H_i(x_n) \| \leq (2 - \left(\frac{1}{2}\right)^n)B'\delta \]

and so \[ \| I - J_G(x_n)^{-1}A_n \| < 2BB'\delta < \frac{1}{3} \]. Hence \( A_n^{-1} \) exists and is bounded in norm by \( B(1 - 2BB'\delta)^{-1} \), so \( x_{n+1} \) exists.

\[
e_{n+1} \leq \| A_n^{-1} \| \cdot \left[ \| G(x^*) - G(x_n) - J_G(x_n)(x_n - x^*) \| + \| J_G(x_n) - A_n \| e_n \right]
\]

\[
\leq \| A_n^{-1} \| \cdot \left[ \gamma_1 e_n + (2 - \left(\frac{1}{2}\right)^n)B'\delta \right] e_n
\]

\[
\leq B(1 - 2BB'\delta)^{-1}[\left(\frac{1}{2}\right)^nB'\delta + (2 - \left(\frac{1}{2}\right)^n)B'\delta] e_n
\]

\[
\leq 2BB'\delta(1 - 2BB'\delta)^{-1}e_n
\]

\[
\leq \frac{1}{2} e_n
\]

Now from Lemma 1,

\[
\| B_{i,n+1} - H_i(x_{n+1}) \| \leq \| B_{i,n} - H_i(x_n) \| + 3K(e_n + e_{n+1})
\]

\[
\leq (2 - \left(\frac{1}{2}\right)^n)\delta + \frac{9}{2}Ke_n
\]

\[
\leq [2 - \left(\frac{1}{2}\right)^n + \frac{1}{2}(\frac{1}{2})^n]\delta
\]

\[
= [2 - (\frac{1}{2})^{n+1}]\delta
\]
and the induction is complete. This implies that the sequence exists
and $e_n \leq \left(\frac{1}{2}\right)^ne_0 \rightarrow 0$.

**THEOREM 2.** If the hypotheses of Theorem 1 hold and $||F(x^*)|| = 0$, then the iteration defined by Algorithm 2.1 or 2.2 converges at least quadratically.

**Proof.**

$e_{n+1} \leq B(1 - 2BB'\delta)^{-1}[\gamma_1e_n^2 + ||F(x_n)||_1(2 - \left(\frac{1}{2}\right)^n\delta e_n]].$

Now $||F(x_n)||_1 = ||F(x_n) - F(x^*)||_1 \leq \max_{x \in C} ||J_F(x)||e_n \equiv B''e_n.$

Hence, $e_{n+1} \leq B(1 - 2BB'\delta)^{-1}[\gamma_1 + 2B''\delta]e_n^2.$

If we assume the stronger continuity condition L2 for the $H_i$, namely

$||H_i(x) - H_i(y)|| \leq K_i||x - y||$, $i = 1, \ldots, M$,

then it is not even necessary to assume the existence of a zero, $x^*$, of $G$; that is, by making assumptions about the behavior of the function and its derivatives in an open convex subset of $E^N$ we are able to prove a Kantorovitch-type theorem [13] for the iterations defined by Algorithms 2.1 and 2.2 in which the existence of $x^*$ is deduced as a part of the proof.
THEOREM 3. Let the following conditions hold in \( \overline{N}(x_o, r) \).

i) \( J_G \) satisfies L2 with constant \( Y \).

ii) Let \( B_{1,0}, \ldots, B_{M,0} \) be real \( N \times N \) matrices and \( \delta > 0 \) such that

\[
\left\| \sum_{i=1}^{M} f_i(x_o) \left[ B_{i,0} - H_i(x_o) \right] \right\|_2 \leq \delta.
\]

iii) \( A_o = \sum_{i=1}^{M} f_i(x_o) B_{i,0} + J_F(x_o)^T J_F(x_o) \)

is invertible and its inverse is bounded in norm by \( \beta \).

iv) \( \left\| A_o^{-1} J_F(x_o)^T F(x_o) \right\|_2 = \left\| x_1 - x_o \right\|_2 \leq \eta \).

Under conditions (i) - (iv), if \( 1 > \beta \delta, \frac{1}{2} \geq h' = \beta \gamma (1 - \beta \delta)^{-2} \)

and \( r \geq r'_o = (1 - \sqrt{1 - 2h'}) (1 - \beta \delta) (\beta \gamma)^{-1} \),

then \( \| F(\cdot) \|_2 \) has a unique minimum at \( x^* \in \overline{N}(x_o, r'_o) \) and if \( h' < \frac{1}{2}, x^* \) is also unique in \( \overline{N}(x_o, r) \cap N(x_o, r_1) \),

where \( r_1 = (1 + \sqrt{1 - 2h'}) (1 - \beta \delta) (\beta \gamma)^{-1} \).

Furthermore, the sequence \( x'_{n+1} = x'_n - A_o^{-1} F(x'_n) \) remains in \( N(x_o, r'_o) \) and converges linearly to \( x^* \).
Proof. Let \( x \in N(x_0, r'_0) \) and
\[
||A_0 - J_G(x)||_2 \leq ||A_0 - J_G(x_0)||_2 + ||J_G(x_0) - J_G(x)||_2 \leq \delta + \gamma ||x_0 - x||_2.
\]
The theorem now follows from Theorem 2 of [8] applied to \( G \).

The last theorem constitutes the first half of a Kantorovich-type theorem [13,18], i.e., the existence and uniqueness of \( x^* \). The other half, the convergence of the algorithms 2.1 and 2.2 to \( x^* \), is contained in the following theorem.

**THEOREM 4.** Let conditions (i) – (iv) hold as well as the following conditions.

v) For every \( i = 1, \ldots, M \), \( H_i \) satisfies L2 with constant \( K \).

vi) \( B'' \geq ||F(x)||_1 \) uniformly for \( r \geq ||x - x_0||_2 \).

vii) For every \( i = 1, \ldots, M \), \( ||H_i(x_0) - B_{i,0}||_2 \leq \delta' \).

Under conditions (i) – (vii), if
\[
1 > \beta \Delta \equiv \beta (2B'' \delta' + \delta), \quad \frac{1}{2} > h \equiv \frac{(3B''K + \gamma) \beta n}{(1 - \beta \Delta)^2},
\]
and \( r \geq r_0 \equiv (1 - \sqrt{1 - 2h}) (1 - \beta \Delta) / \beta (3B''K + \gamma) \).
NEW ALGORITHMS FOR MINIMIZING A SUM OF SQUARES

then

1) $F$ has a root $x^*$ in $\mathbb{N}(x_0, r_0')$.

2) Algorithms 2.1 and 2.2 converge to $x^*$.

3) $x^*$ has the uniqueness properties ascribed to it in Theorem 3.

Proof. The proof will be similar to the proof of Theorem 3 of [8] and will be based on the techniques of [9]. As in Theorem 1 we will use the bounds of Lemma 2, since the bounds for Algorithm 2.2 also hold for Algorithm 1.1.

Since $\Delta \geq \delta$, $h \geq h'$ and $r_o \geq r_0'$ we can deduce the existence and uniqueness of $x^*$ from the previous theorem.

Assume, for $n \geq 0$, that

$$ \sum_{i=0}^{n} ||x_{i+1} - x_i||_2 < r_o. $$

Then $J_G(x_{n+1})$, $F(x_{n+1})$ and $B_{i,n+1}$, $i = 1, \ldots, M$, are all defined.

Hence $A_{n+1}$ is defined and, proceeding as in the proof of Theorem 1,

$$ ||A_{n+1} - J_G(x_{n+1})||_2 = \left| \sum_{i=1}^{M} f_i(x_{n+1}) [B_{i,n+1} - H_i(x_{n+1})] \right|_2 $$

$$ \leq \left| ||F(x_{n+1})||_1 \max_i ||B_{i,n+1} - H_i(x_{n+1})||_2 \right|. $$

We can use induction in connection with Lemma 2 and use (v) to obtain, for every $i = 1, \ldots, M$, 

\[ ||B_{i,n+1} - H_{i}(x_{n+1})||_2 \leq \delta' + 2K \sum_{i=0}^{n} ||x_{i+1} - x_i||_2.\]

Hence,
\[ ||A_{n+1} - J_G(x_{n+1})||_2 \leq B''\delta' + 2B''K \sum_{i=0}^{n} ||x_{i+1} - x_i||_2\]

and so
\[ ||A_0^{-1} A_{n+1} - I||_2 \leq \beta ||A_{n+1} - J_G(x_{n+1})||_2 + ||J_G(x_{n+1}) - J_G(x_o)||_2\]
\[ = \beta \sum_{i=0}^{n} ||x_{i+1} - x_i||_2 + \gamma ||x_{n+1} - x_0||_2 + \beta \Delta + (2B''K + \gamma) \beta \sum_{i=0}^{n} ||x_{i+1} - x_i||_2 + \beta \Delta + (3B''K + \gamma) \beta r_0\]
\[ = \beta \Delta + (1 - \beta \Delta) (1 - \sqrt{1 - 2\eta}) \]
\[ \leq 1.\]

Hence \( A_{n+1}^{-1} \) exists by the Banach Lemma [13,18] and is bounded in norm by
\[ (1 - \beta \Delta - (2B''K + \gamma) \beta \sum_{i=0}^{n} ||x_{i+1} - x_i||_2) .\]
Let \( a_o = \beta^{-1}, \ t_o = 0, \)

\[
f(t) = \frac{1}{2} (3B''K + \gamma)t^2 - (\beta^{-1} - \Delta)t + \beta^{-1} \eta,
\]

\[
a_k = \beta^{-1}(1 - \beta B''\delta' - \beta \delta - \beta(2B''K + \gamma)t_k), \ k > 0
\]

and consider the sequence

\[
t_{k+1} = t_k - a_k^{-1} f(t_k).
\]

(The functions \( a(t) \) and \( f(t) \) were found by the methods of Theorem 2 of [9] and the convergence conditions were obtained from Theorem 3 of [9].) The conditions on \( r_o, h, \) etc. ensure that the sequence \( \{t_k\} \) defined above is monotone increasing and converges to \( r_o \), which is a zero of \( f \). The conditions (i) - (vii) ensure that \( ||x_{k+1} - x_k||_2 \leq t_{k+1} - t_k \) and so by a standard argument \( \{x_k\} \) must converge. It is fairly straightforward to show that its limit is \( x^* \). Details of the proof can be found in [9] or, with different parameters, in [8].

**REMARK 3.2.** If we proceed as in Lemma 2, we can reduce the number of parameters in the previous two results. This is accomplished by keeping (v), deriving
\[ \|J_F(x)\|_2 \leq \|J_F(x_o)\|_2 + MK \|x - x_o\|_2 \] and

\[ \|F(x)\|_2 \leq \|F(x_o)\|_2 + \|J_F(x_o)\|_2 \cdot \|x - x_o\|_2 + MK \|x - x_o\|_2^2 \]

and replacing (i) by means of Lemma 3.

We also note that the proof techniques of this section apply to any algorithms of the types considered in this paper which use a sequence \(B_{i,n}\), where

\[ \|B_{i,n} - H_i(x_n)\|_2 \leq \delta_o + \delta_1 \sum_{j=1}^{n} \|x_j - x_{j-1}\|_2 \]
4. Numerical Results. In the examples which follow, we use the term equivalent function evaluations to mean the total of each evaluation of the function \( F \) and each evaluation of a column of the Jacobian matrix \( J_F \); note that for the examples studied, it takes less computational effort to evaluate a Jacobian column than it does to evaluate a component of the gradient function \( \nabla \psi \).

Examples 1 and 2 were run in FORTRAN IV double precision (\( \approx 16 \) decimal digits) on Cornell University's IBM 360/65. Example 3 was run in FORTRAN IV single precision (\( \approx 8 \) decimal digits) on the Yale University Department of Computer Science's PDP-10.

Example 4.1. A "classical" example. In order to test the methods against a variety of algorithms in current use, we referred to the very fine survey paper by Box [3]. The test function used was

\[
\psi(x_1, x_2, x_3) = \sum_{i=1}^{10} \left[ e^{-x_1 p_i} - e^{-x_2 p_i} \right] - x_3(e^{-p_1} - e^{-10p_1}) \]

where \( p_i = .1 \times i \). This problem has a zero residual at \((1, 10, 1)\) and whenever \( x_1 = x_2 \) with \( x_3 = 0 \). We used those starting points for which \( \psi(x^{(0)}) \) was large:

I. \( x_1 = 0, \ x_2 = 10, \ x_3 = 20; \ psi = 1031.154 \)

II. \( x_1 = 0, \ x_2 = 20, \ x_3 = 20; \ psi = 1021.655 \).

The results are presented in the Table.
<table>
<thead>
<tr>
<th>Method</th>
<th>Starting Point I</th>
<th>Starting Point II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swann [24]</td>
<td>Failed</td>
<td>Failed</td>
</tr>
<tr>
<td>Rosenbrock [22]</td>
<td>350</td>
<td>246</td>
</tr>
<tr>
<td>Nelder and Mead [16]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spendley, Hext and Himsworth [23]</td>
<td>307</td>
<td>315</td>
</tr>
<tr>
<td>Powell (1964) [19]</td>
<td>Failed</td>
<td>Failed</td>
</tr>
<tr>
<td>Fletcher and Reeves [11]</td>
<td>92</td>
<td>188</td>
</tr>
<tr>
<td>Davidon [7]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fletcher and Powell [10]</td>
<td>140</td>
<td>140</td>
</tr>
<tr>
<td>Powell (1965) [20]</td>
<td>28</td>
<td>33</td>
</tr>
<tr>
<td>Barnes [2]</td>
<td>37</td>
<td>59</td>
</tr>
<tr>
<td>Gauss [4]</td>
<td>17</td>
<td>21</td>
</tr>
<tr>
<td>Derivative-free Gauss [4]</td>
<td>17</td>
<td>21</td>
</tr>
<tr>
<td>Levenberg-Marquardt [4]</td>
<td>93</td>
<td>109</td>
</tr>
<tr>
<td>Algorithm 2.1 including evaluations done to approximate $H_1(x^{(o)})$</td>
<td>30</td>
<td>34</td>
</tr>
<tr>
<td>Algorithm 2.1 when $H_1(x^{(o)})$ was given</td>
<td>24</td>
<td>28</td>
</tr>
<tr>
<td>Algorithm 2.2 including evaluations done to approximate $H_1(x^{o})$</td>
<td>30</td>
<td>34</td>
</tr>
<tr>
<td>Algorithm 2.1 when $H_1(x^{o})$ was given</td>
<td>24</td>
<td>28</td>
</tr>
</tbody>
</table>
REMARK 4.1. Many of the methods above behave linearly and could not be expected to rapidly reduce $\phi$ from $10^{-5}$ to $10^{-10}$; however, Algorithms 2.1 and 2.2 exhibited quadratic convergence in this range.

Example 4.2. Weights and nodes of quadrature rules. This example is given by Nielsen [17] and is an illustration of how quadrature weights and nodes may be calculated by nonlinear least squares techniques.

Data Vectors:

<table>
<thead>
<tr>
<th>P</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>2.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2.0</td>
<td>2/3</td>
</tr>
<tr>
<td>3.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4.0</td>
<td>2/5</td>
</tr>
<tr>
<td>5.0</td>
<td>0.0</td>
</tr>
<tr>
<td>6.0</td>
<td>2/7</td>
</tr>
<tr>
<td>7.0</td>
<td>0.0</td>
</tr>
<tr>
<td>8.0</td>
<td>2/9</td>
</tr>
<tr>
<td>9.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Functional Relationship: $g(x; p) = x_1 x_3^p + x_2 x_4^p$.

Initial Approximation: $x_1 = 1.0$, $x_2 = 1.0$, $x_3 = -.75$, $x_4 = .75$.

Both Algorithms 2.1 and 2.2 achieved convergence in 8 iterations to ten significant digits of accuracy to the minimum, $x^* = (.97754, .97754, -.65140, .65140)$, whereas the Gauss-Newton method [17, p. 41] required 8 iterations to achieve five digits of accuracy. (Note that equivalent function evaluations per iteration are the same for the methods compared.)
Example 4.3. A problem with a very large residual at the minimum.

Let

\[ p_i = .2 \times i, \quad i = 1, \ldots, 20 \]

and suppose we wish to minimize

\[
\phi(x) = \sum_{i=1}^{20} \left( x_1 + x_2 p_i - e_i p_i \right)^2 + \left( x_3 + x_4 \sin p_i - \cos p_i \right)^2.
\]

Using \( x^{(0)} = (25, 5, -5, -1) \), with the initial residual being \( \phi(x^{(0)}) = 7926693 \), Algorithm 2.1 converged in 7 iterations — requiring 50 equivalent function evaluations — to the minimum \( x^* = (-11.594, 13.204, -.40344, .23678) \). The residual at the minimum was \( \phi(x^*) = 85822 \) and \( G(x^*) \approx 85822 \). Note that the total of 50 evaluations included 10 necessary to approximate the initial (symmetric) Hessians, \( H_k(x^{(0)}) \), \( k = 1, \ldots, M \).

Using the same starting guess, Algorithm 2.2 converged to the same solution point, also requiring 7 iterations and 50 equivalent function evaluations.
REFERENCES


